

University of Bahrain  
College of Information Technology  
Department of Computer Science  
Semester 1, 2013-2014  
ITCS216 (Data Structures and Algorithms)

**Final Exam**

**Date:** January 4, 2014

**Time:** 11:30-13:30

STUDENT NAME	.....
STUDENT ID #	.....

**NOTES:**

- WRITE ONLY ONE SOLUTION FOR EACH QUESTION.
- SWITCH OFF YOUR MOBILE PHONES.
- THIS EXAM CONTAINS 9 PAGES.

QUESTION #	MARKS		COMMENTS
1	8		
2	18		
3	8		
4	6		
TOTAL	40		

### **Question 1: Sorting Algorithms (3-5 marks)**

<b>Data:</b>	1	18	5	11	4	6	0	25	17	10	3	21	11	13	0	29	31	19	20	8	7
--------------	---	----	---	----	---	---	---	----	----	----	---	----	----	----	---	----	----	----	----	---	---

The objective of this question is to sort the given array using the Shell sort algorithm.

1. Use the procedure shown in the lecture to determine the numbers  $h_t \dots h_1$  of ways of dividing the given array into subarrays.

- For each iteration of the Shell sort algorithm, give the elements of the different subarrays before being sorted and the elements of the array data after sorting all the subarrays.

## **Question 2: Linked lists & Queues (18 marks)**

In an emergency unit of a hospital, patients are placed in a waiting queue and assigned a priority based on their health state. The more the case is critical, higher is its priority. The subject of this question is to design a priority queue to manage the patients using an ordered linked list.

1. Design a java class named **patient** which will represent a node in the ordered linked list. The class patient has three members; the 1<sup>st</sup> to store the name of the patient as a string, the 2<sup>nd</sup> to assign a priority (integer), and the 3<sup>rd</sup> to link the current patient to the next patient. Add a constructor with three parameters to the class patient **(3 marks)**.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Design a java class named **orderedLinkedList** with the following members **(4 marks)**:
  - Two **protected** members, head and tail to point the first and the last nodes respectively.
  - A public constructor to initialize the ordered linked list to empty.
  - A public method **isEmpty** to indicate whether the ordered linked list is empty.
  - A public method **addPatient** which takes two parameters; the name of the patient and its priority and inserts it in the linked list. After the insertion of the new patient, the list must remain sorted in a decreasing order of the priorities (The patient having the highest priority is placed in the head of the list). We suppose that all the patients have different priorities.
  - A public method **delete** which deletes the patient having the highest priority from the list.
  - A public method **print** to print the list of patients and their priorities according to the following way:

Patient1: Name1, priority1  
Patient2: Name2, priority2

.

.

.

3. Implement the constructor and all the methods of the class **orderedLinkedList** (6 marks).

Page 4/9





### Question 3: Binary Trees & Stacks (4-4 marks)

1. Write a member method `addLeaves` to be added to the class `binaryTree`. The method `addLeaves` adds and returns the info of the leaves of the binary tree pointed by a node-type parameter `p`.

.....

.....

.....

.....

.....

.....

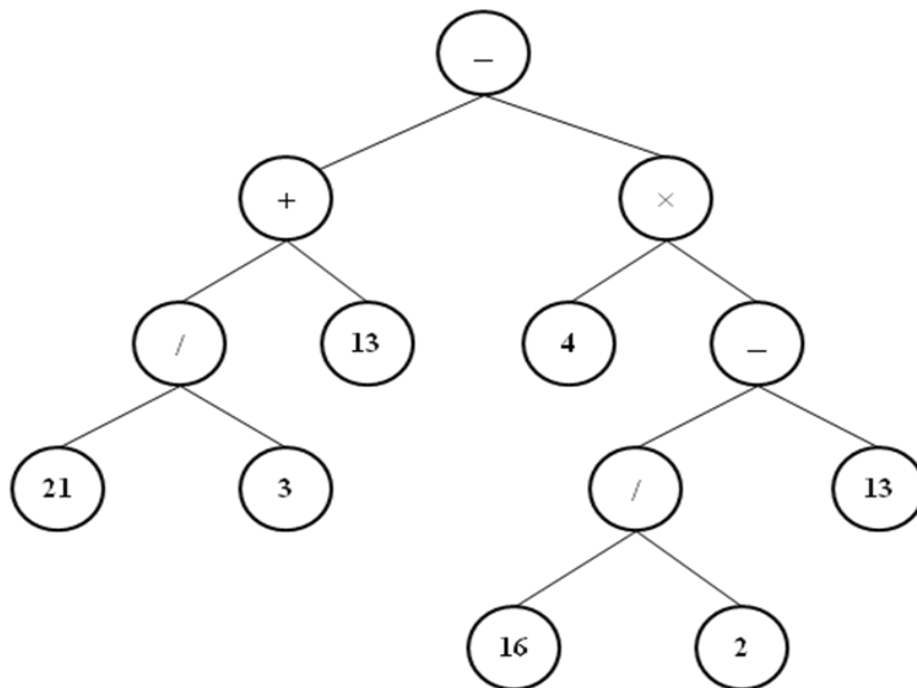
.....

.....

.....

.....

2. Show the result of applying a post-order traversal to the following binary tree.



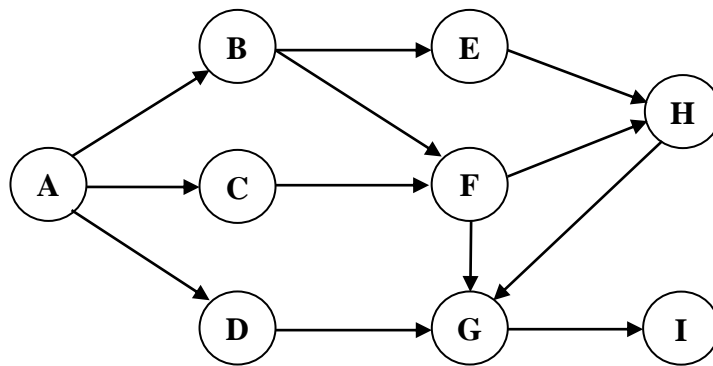
.....

.....





### Question 4: Graphs (3-3 marks)



1. Find the adjacency list representation of the above directed graph

2. Use a depth-first traversal to visit the vertices of the given directed graph starting from A.